Interpolatory model order reduction by tensor Krylov methods

Pieter Lietaert *Karl Meerbergen*

Report TW 659, May 2015

KU Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Interpolatory model order reduction by tensor Krylov methods

Pieter Lietaert *Karl Meerbergen*

Report TW 659, May 2015

Department of Computer Science, KU Leuven

Abstract

High dimensional models with parametric dependencies can be challenging to simulate. The computational effort usually increases exponentially with the dimension of the parameter space. To keep the calculations feasible, one can use parametric model order reduction techniques. Multivariate Pad methods match higher order moments of the Laplace variable as well as the parameters. Interpolatory reduced models interpolate the exact transfer function for different parameter values. In this paper we use tensor Krylov techniques to reduce the parametric model, combining both moment matching and interpolatory model reduction in the parameter space. If a low rank tensor formulation is possible, this approach is competitive with classic parametric model reduction. Furthermore, we look at models containing stochastic parameters and construct a model that outputs the mean over these parameters within the domain of interest. This model for the mean is set up using well known quadrature techniques from numerical integration. We compare the reduced model for the mean with the parametric reduced model.

Keywords : Model order reduction, tensors, Krylov methods, moment matching, stochastic parameters
MSC : Primary : 15A69, 15A30, 15B51, 65F99, 65F10

INTERPOLATORY MODEL ORDER REDUCTION BY TENSOR KRYLOV METHODS

KARL MEERBERGEN, PIETER LIETAERT

Abstract. High dimensional models with parametric dependencies can be challenging to simulate. The computational effort usually increases exponentially with the dimension of the parameter space. To keep the calculations feasible, one can use parametric model order reduction techniques. Multivariate Padé methods match higher order moments of the Laplace variable as well as the parameters. Interpolatory reduced models interpolate the exact transfer function for different parameter values. In this paper we use tensor Krylov techniques to reduce the parametric model, combining both moment matching and interpolatory model reduction in the parameter space. If a low rank tensor formulation is possible, this approach is competitive with classic parametric model reduction. Furthermore, we look at models containing stochastic parameters and construct a model that outputs the mean over these parameters within the domain of interest. This model for the mean is set up using well known quadrature techniques from numerical integration. We compare the reduced model for the mean with the parametric reduced model.

1. Introduction. Consider the following descriptor system

$$\begin{aligned} E\dot{x} + Ax &= fu(t) \\ y &= c^T x \end{aligned} \tag{1.1}$$

where $A, E \in \mathbb{R}^{n\times n}$ and $f, c \in \mathbb{R}^n$ with large n. In order to reduce the complexity for computing y, we use model order reduction techniques. Model reduction has become a standard tool in many disciplines in science and engineering, e.g., for the design of electronic devices, civil constructions, mechanical engineering, including challenging problems such as flexible multibody dynamics and contact problems. Model reduction techniques for linear and nonlinear SISO systems, described in state space form, is understood. For linear models, represented in the Laplace or frequency domain, we identify several classes of methods: moment matching through (rational) Krylov methods, e.g., [14], [18], [36], and [5]; balanced truncation, e.g., [32], [1], [3], [4], [21], [7], and [19]; the iterative rational Krylov method for $\mathcal{H}_2$ error norm minimization, [19][20]; and the dominant pole algorithm [33].

In this paper, matrices A and E depend on d uncorrelated stochastic parameters, denoted by vector $\gamma \in \Gamma = [a_1, b_1] \times \cdots \times [a_d, b_d] \subset \mathbb{R}^d$ with d a moderate number. We are interested in the mean behaviour of system (1.1) over the parameter space, i.e., we want to know

$$z = \int_\Gamma y\phi(\gamma)d\gamma$$

where $\phi(\gamma)$ is the distribution of γ over the parameter domain Γ. Since A and E depend on parameters, we use parametric model order reduction techniques. Multivariate Padé methods match the higher order partial derivatives for all variables (Laplace variable and parameters) evaluated at the given interpolation point(s); see, e.g., [10], [38], [27], [28], [26], [22], [17], [15], [25], [29], and [16]. These methods are generic, i.e. they can be applied to many problems, usually with affine parameters. Interpolatory reduced models interpolate the exact transfer function in interpolation points for the parameters and the Laplace variable [2]. The last few years, the research focus was on the choice of interpolation points. The choice of these points is by a great deal inspired by the numerical integration community, which has acquired significant expertise on approximation of functions in many variables, e.g., sparse grids in [31], and lattice rules in [35]. The interpolation points can also be chosen adaptively, using an error estimation. This leads to the class of Reduced Basis methods, which have been developed over the last years [30][34]. The points are chosen by solving an optimization problem, e.g., through a greedy approach. An a-posteriori error estimation drives this selection. There is a vast literature on this topic. Also see recent work on the selection of interpolation points in the Laplace domain [8].

The goal of this paper is twofold. First, the computation of z by numerical integration may require many solutions to the system, which is expensive, even when a reduced model is used. We

will therefore attempt to build a reduced model for the mean itself, i.e., develop a SISO system that does not contain parameters, whose input is $u(t)$ and whose output is an approximation to z obtained by numerical integration. The second and main objective of the paper is to use a tensor representation of x and y by discretization of Γ in a Cartesian grid. This does not sound an appealing choice due to the exponential increase of the number of points with the number of parameters, but we expect a significant reduction of the cost by using low rank tensors. In this paper, we will only concentrate on moment matching methods with interpolation points for the Laplace variable at zero or another finite shift. The results are easily extended to rational Krylov methods. In order to simplify the numerical methods and complexity, we assume that A and E are affine functions of γ and that γ is a vector of statistically uncorrelated parameters, i.e., ϕ can be factored as

$$\phi(\gamma) = \phi_1(\gamma_1) \cdots \phi_d(\gamma_d).$$

1.1. Outline of the paper. The paper is organized as follows. Section 2 presents a SISO system whose output is an approximation of the mean, z. We introduce tensor-Krylov methods as in [23], but now for model reduction. In each iteration of the tensor-Krylov method, a linear system parametrized in γ, needs to be solved. We exploit low rank structures of the tensors and use tensor-Krylov methods from [23][6]. In section 3 we analyse the moment matching properties for the SISO system with output z. We explain how to choose the interpolation points in the parameter space. A connection with interpolatory model reduction and multivariate moment matching is shown when a particular preconditioner and Krylov solver is used. A practical two-level algorithm is also presented. Section 5 illustrates the algorithms by numerical examples. We conclude the paper with final comments and suggestions for future work in §6.

1.2. Notation and tensors. We will denote by $\gamma \in \mathbb{R}^d$ the vector of parameters. The j-th parameter is denoted by γ_j. For other variables, lower case Roman characters denote vectors, upper case characters denote matrices. We use $\mathbf{1}_n$ as a vector of length n containing all ones. Tensors are denoted using calligraphic letters, e.g., $\mathcal{X}$. The numbering of the modes of tensors will begin at zero. This is very unusual in the tensor community, but we find it easier in this paper to denote the tensors and tensor operations with this convention. The notation for operations on tensors is taken from [12] and [13].

The rank-one tensor $\mathcal{X}$, formed by the outer product of the vectors $x_0, \ldots, x_d$, is denoted by

$$\mathcal{X} = (x_0, \ldots, x_d).$$

The rank of a tensor $\mathcal{A}$ is defined as the minimum value R so that $\mathcal{A}$ can be written as the sum of R rank-one tensors:

$$\mathcal{A} = \sum_{j=1}^{R} (x_0^{(j)}, \ldots, x_d^{(j)}).$$

The multilinear matrix multiplication of tensor $\mathcal{A}$ with matrices $B_0, B_1, \ldots, B_d$ is denoted as:

$$\mathcal{A}' = (B_0, B_1, \ldots, B_d) \cdot \mathcal{A}.$$

The former product is the multiplication of a tensor in all modes. The notation is extended to multiplication in a restricted number of modes by writing down these modes as subscripts, e.g.

$$\mathcal{A}' = (B_3, B_4) \cdot_{3,4} \mathcal{A}.$$

The inner product of two tensors with the same dimensions is denoted as $\langle \mathcal{A}, \mathcal{B} \rangle$ and defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_0, \ldots, i_d} \overline{\mathcal{A}}_{i_0, \ldots, i_d} \cdot \mathcal{B}_{i_0, \ldots, i_d}, \tag{1.2}$$

which is the equivalent of the trace of $\mathcal{A}^* \mathcal{B}$ when both tensors are matrices. The induced norm is denoted by $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

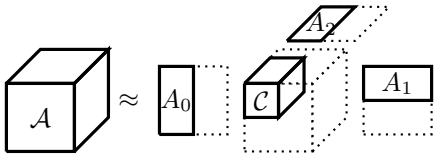

Fig. 1.1: Graphical representation of the Tucker decomposition of a tensor of order three.

Fibers are the extension of rows and columns of a matrix to tensors. A j-fiber is obtained by fixing all indices except index j. In this paper, we use the following notation to indicate a mode-j fiber:

$$\mathcal{A}_{i_0,i_1,\ldots,i_{j-1},:,i_{j+1},\ldots,i_d} = (e_{i_0}, e_{i_1}, \ldots, e_{i_{j-1}}, e_{i_{j+1}}, \ldots, e_{i_d}) \cdot_{0,\ldots,j-1,j+1,\ldots,d} \mathcal{A}.$$

In the matrix case, the mode-0 fibers correspond to columns and the mode-1 fibers to rows. We define the mode-j vector space of $\mathcal{A}$ as the space spanned by all mode-j fibers. In the matrix case, the mode-0 vector space is the column space and the mode-1 space is the row space.

The ith slice of a tensor in the jth direction is

$$\mathcal{B}_{i_0,\ldots,i_{d-1}} = \mathcal{A}_{i_0,\ldots,i_{j-1},i,i_{j+1},\ldots,i_d},$$

i.e., $\mathcal{B}$ contains all elements of $\mathcal{A}$ where the j-coordinate is fixed to i. A slice of an order 3 tensor is a matrix. We denote the ith slice in the jth direction as

$$\mathcal{A}_{\underbrace{:,\ldots,:}_{j\times},i,\underbrace{:,\ldots,:}_{d-j\times}}.$$

In what follows, we will presume that we can use low rank formulations to represent the tensors. Specifically, the Tucker format is used. By a Tucker representation we mean:

$$\mathcal{A} = (A_0, \ldots, A_d) \cdot \mathcal{C},$$

with tensor $\mathcal{A} \in \mathbb{C}^{n_0 \times \cdots \times n_d}$, factor matrices $A_j \in \mathbb{C}^{n_j \times r_j}$ and core tensor $\mathcal{C}^{r_0 \times \cdots \times r_d}$. When $r_j \ll n_j$, for some $j \in \{0, \ldots, d\}$, $\mathcal{A}$ is a low rank tensor and its Tucker representation requires by far less storage. Figure 1.1 shows a graphical representation of a Tucker decomposition. The best known algorithm for computing a Tucker decomposition from a tensor is the HOSVD method [11]. When $\mathcal{A}$ is a matrix, the Tucker decomposition corresponds to:

$$(A_0, A_1) \cdot \mathcal{C} = A_0^* \mathcal{C} A_1.$$

The Tucker format can thus be seen as an extension of the singular value decomposition of a matrix where the core tensor, in every case, is a diagonal matrix.

2. A tensor Krylov method for the mean. In this section, we present a single-input-single-output system (SISO) whose output is a quadrature approximation of the mean of (1.1). We propose a tensor Krylov method for reducing this model.

2.1. Restriction to the parameter grid. Let $v : \Gamma \to \mathbb{C}^n$ be a multivariate function. Recall that $\Gamma = [a_1, b_1] \times \cdots \times [a_d, b_d]$. Let $[a_j, b_j]$ be discretized by points $\gamma_j^{(1)}, \ldots, \gamma_j^{(N_j)}$ for $j = 1, \ldots, d$. These points form a grid, $\mathbb{G}$, with

$$\mathbb{I} = \{(i_1, \ldots, i_d), i_j = 1, \ldots, N_j, j = 1, \ldots, d\},$$
$$\mathbb{G} = \{\gamma^{(I)} = (\gamma_1^{(i_1)}, \ldots, \gamma_d^{(i_d)}), I = (i_1, \ldots, i_d) \in \mathbb{I}\}$$

and $N = N_1 N_2 \cdots N_d$ points. We discretize $v : \Gamma \to \mathbb{C}^n$ by its restriction to the grid points and represent it by the order $d + 1$ tensor:

$$\mathcal{V} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d} : \mathcal{V}_{:,i_1,\ldots,i_d} = v(\gamma^{(i_1,\ldots,i_d)}) \quad , \quad (i_1, \ldots, i_d) \in \mathbb{I},$$

i.e., each mode-0 fiber corresponds to the evaluation of v in a grid point. We will often use the following notation to address mode-0 fibers:

$$\mathcal{V}_{:,J} = v(\gamma^{(J)}) \quad , \quad J \in \mathbb{I}.$$

We assume that A and E in (1.1) are affine in γ, i.e., there are A_j and E_j so that

$$A(\gamma) = A_0 + \sum_{j=1}^{d} \gamma_j A_j,$$
$$E(\gamma) = E_0 + \sum_{j=1}^{d} \gamma_j E_j.$$

In this paper, these matrices are often applied on a vector $x \in \mathbb{C}^n$ that is a function of γ. Since x is represented by tensor $\mathcal{X} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d}$, we replace A and E by operators $\mathcal{A}$ and $\mathcal{E}$ to denote the action of A and E for $\gamma \in \mathbb{G}$. Operator $\mathcal{A}$, respectively $\mathcal{E}$, maps $\mathcal{X}$ to $\mathcal{A}(\mathcal{X})$, respectively $\mathcal{E}(\mathcal{X}) \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d}$, defined by their mode-0 fibers:

$$\begin{aligned} \mathcal{A}(\mathcal{X})_{:,J} &:= \quad (A_0 + \sum_{j=1}^{d} \gamma_j^{(i_j)} A_j)\mathcal{X}_{:,J} \quad , \quad J = (i_1, \ldots, i_d) \in \mathbb{I}, \\ \mathcal{E}(\mathcal{X})_{:,J} &:= \quad (E_0 + \sum_{j=1}^{d} \gamma_j^{(i_j)} E_j)\mathcal{X}_{:,J} \quad , \quad J = (i_1, \ldots, i_d) \in \mathbb{I}, \end{aligned} \tag{2.1}$$

i.e., $\mathcal{A}$ and $\mathcal{E}$ are discretizations of $A(\gamma)$ and $E(\gamma)$ with $\gamma \in \Gamma$. In tensor notation, these operators can be written as

$$\mathcal{X} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d} \mapsto \mathcal{A}(\mathcal{X}) = (A_0)\cdot_0 \mathcal{X} + \sum_{j=1}^{d} (A_j, \Sigma_j)\cdot_{0,j} \mathcal{X} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d},$$
$$\mathcal{X} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d} \mapsto \mathcal{E}(\mathcal{X}) = (E_0)\cdot_0 \mathcal{X} + \sum_{j=1}^{d} (E_j, \Sigma_j)\cdot_{0,j} \mathcal{X} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d},$$

where $\Sigma_j = \mathrm{diag}(\gamma_j^{(1)}, \ldots, \gamma_j^{(N_j)})$. We will mostly use the notation in (2.1). Operators $\mathcal{A}$ and $\mathcal{E}$ are linear.

In order to compute the mean of $y : \Gamma \to \mathbb{C}$, by numerical integration, we use a quadrature rule for each parameter, i.e., for each $j = 1, \ldots, d$, we choose abscissae $\gamma_j^{(i)}$ and associated weights $w_j^{(i_j)}$, $i_j = 1, \ldots, N_j$ so that

$$z = \int_\Gamma y\phi_1 \cdots \phi_d \approx z_N = \sum_{i_1=1}^{N_1} \cdots \sum_{i_d=1}^{N_d} y(\gamma^{(i_1,\ldots,i_d)}) w_1^{(i_1)} \cdots w_d^{(i_d)}.$$

Note that all elements of w_j are non-zero and all interpolation points are distinct as both arise from a (classical) quadrature rule. When $\mathcal{Y} \in \mathbb{C}^{N_1 \times \cdots \times N_d}$ is the tensor representation of the discretization of y, then this summation can be written as the following tensor contraction

$$z_N = (w_1^T, \ldots, w_d^T)\cdot \mathcal{Y}.$$

2.2. A SISO system for the mean. Consider the following SISO system:

$$\begin{aligned} s\mathbf{E}\mathbf{x}(s) + \mathbf{A}\mathbf{x}(s) &= \mathbf{f}u(s) \\ z_N &= (w_d^T \otimes \cdots \otimes w_1^T \otimes c^T)\mathbf{x}(s) \end{aligned} \tag{2.2}$$

with

$$\mathbf{E} = \begin{bmatrix} E(\gamma^{(1,\ldots,1)}) & & \\ & \ddots & \\ & & E(\gamma^{(N_1,\ldots,N_d)}) \end{bmatrix} \quad , \quad \mathbf{A} = \begin{bmatrix} A(\gamma^{(1,\ldots,1)}) & & \\ & \ddots & \\ & & A(\gamma^{(N_1,\ldots,N_d)}) \end{bmatrix}$$

$$\mathbf{x} = \begin{pmatrix} x_{1,\ldots,1}(s) \\ \vdots \\ x_{N_1,\ldots,N_d}(s) \end{pmatrix} \quad , \quad \mathbf{f} = \mathbf{1}_N \otimes f,$$

with s the Laplace variable. The output is the approximation of the mean z. To reduce this system, we consider two-sided Krylov methods for model reduction. The results can easily be extended to rational Krylov methods. We will mention this in the theoretical results. The following algorithm sketches the main idea.

ALGORITHM 2.1 (Moment matching for (2.2)).

1: Choose an interpolation point σ
2: Build an orthonormal basis $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ of the Krylov sequence

$$\{(\sigma\mathbf{E} + \mathbf{A})^{-1}\mathbf{f}, \ldots, ((\sigma\mathbf{E} + \mathbf{A})^{-1}\mathbf{E})^{k-1}(\sigma\mathbf{E} + \mathbf{A})^{-1}\mathbf{f}\}$$

3: Set the initial vector $\mathbf{v}_1$:

$$\mathbf{t}_0 = (\sigma\mathbf{E} + \mathbf{A})^{-1}\mathbf{f}$$
$$\mathbf{v}_1 = \mathbf{t}_0/\|\mathbf{t}_0\|$$

4: **for** $\ell = 1, \ldots, k-1$ **do**
5: Solve the linear system $(\sigma\mathbf{E} + \mathbf{A})\mathbf{t}_\ell = \mathbf{E}\mathbf{v}_\ell$
6: Orthonormalize $\mathbf{t}_\ell$ against $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$ resulting in the vector $\mathbf{v}_{\ell+1}$.
7: **end for**
8: Similarly build an orthonormal basis $\mathbf{W}_k = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$ of the Krylov space

$$\{(\sigma\mathbf{E} + \mathbf{A})^{-T}\mathbf{c}, \ldots, ((\sigma\mathbf{E} + \mathbf{A})^{-T}\mathbf{E}^T)^{k-1}(\sigma\mathbf{E} + \mathbf{A})^{-T}\mathbf{c}\}$$

9: Build the reduced model:

$$\begin{aligned} s\widehat{\mathbf{E}}\widehat{x}(s) + \widehat{\mathbf{A}}\widehat{x}(s) &= \mathbf{f} \\ \widehat{z} &= \widehat{\mathbf{c}}^*\widehat{x}(s) \end{aligned} \tag{2.3}$$

with

$$\widehat{\mathbf{E}} = \mathbf{W}_k^*\mathbf{E}\mathbf{V}_k \quad , \quad \widehat{\mathbf{A}} = \mathbf{W}_k^*\mathbf{A}\mathbf{V}_k$$
$$\widehat{\mathbf{f}} = \mathbf{W}_k^*\mathbf{f} \quad \text{and} \quad \widehat{\mathbf{c}} = \mathbf{V}_k^*\mathbf{c}.$$

The reduced model (2.3) contains dense matrices, but when its size k is small, it is cheap to evaluate.

Instead of using full vectors, we now use tensors. At first sight, we do not gain anything by replacing N vectors of dimension n by an $n \times N_1 \times \cdots \times N_d$ tensor. However, we do assume that these tensors can be well approximated by low rank tensors. As a result, we can employ the above moment matching method efficiently. A justification for this assumption will be given in §3.

The state vector $\mathbf{x}$ is represented by a tensor, $\mathcal{X} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d}$. We associate mode-0 with the state vector of length n and mode-($j > 0$) with parameter γ_j. As a result, the mode-0 fibers of $\mathcal{X}$ represent the discrete version of x in the grid points in Γ. System (2.2) can thus be written as

$$\begin{aligned} (\mathcal{A} + s\mathcal{E})(\mathcal{X}) &= (fu(s), \mathbf{1}_{N_1}, \ldots, \mathbf{1}_{N_d}) \\ z_N &= (c^T, w_1^T, \ldots, w_d^T) \cdot \mathcal{X}, \end{aligned} \tag{2.4}$$

Value z_N is the result of a tensor contraction. This system is still linear in the Hilbert space $\mathbb{R}^n \times \mathbb{G} \to \mathbb{R}^n$ using the inner product (1.2). This means that we can use Krylov methods for moment matching.

In order to simplify notation, in what follows, we presume that $\sigma = 0$. The presented algorithm can easily be generalized for other frequencies. Algorithm 2.1 is rewritten using tensor notation. The matrix of iterations vectors, $\mathbf{V}_k = [\mathbf{v}_1, ..., \mathbf{v}_k]$, is represented by an order $d+2$ tensor, $\mathcal{V} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d \times k}$. The ℓ-th iteration vector corresponds to $\mathcal{V}_{:,\ldots,:,l}$, i.e., the l-th slice in direction $d+2$.

ALGORITHM 2.2 (Moment matching using tensors).

1: Build an orthonormal basis, represented by tensor $\mathcal{V} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d \times k}$, corresponding to Step 2 in Algorithm 2.1 ($\sigma = 0$).
2: Solve:

$$\mathcal{A}(\mathcal{T}_0) = (f, \mathbf{1}_{N_1}, \ldots, \mathbf{1}_{N_d}). \tag{2.5}$$

3: Normalize:

$$\mathcal{V}_{:,\ldots,:,1} = \mathcal{T}_0 / \|\mathcal{T}_0\|_F.$$

4: **for** $\ell = 1, \ldots, k-1$ **do**
5: Solve:

$$\mathcal{A}(\mathcal{T}_\ell) = \mathcal{E}(\mathcal{V}_{:,\ldots,:,\ell}). \tag{2.6}$$

6: Orthonormalize $\mathcal{T}_\ell$ against $\mathcal{V}_{:,\ldots,:,1:\ell}$ resulting in tensor $\mathcal{V}_{:,\ldots,:,\ell+1}$
7: **end for**
8: Similarly, build an orthonormal basis represented by tensor $\mathcal{W} \in \mathbb{C}^{n \times N_1 \times \cdots \times N_d \times k}$, corresponding to Step 8 in Algorithm 2.1.
9: Build the reduced model:

$$\begin{aligned} s\widehat{\mathbf{E}}\widehat{x}(s) + \widehat{\mathbf{A}}\widehat{x}(s) &= \widehat{\mathbf{f}} \\ \widehat{z} &= \widehat{\mathbf{c}}^* \widehat{x}(s) \end{aligned} \tag{2.7}$$

with

$$\widehat{\mathbf{E}} = \mathcal{W} \cdot_{0,\ldots,d} \mathcal{E}(\mathcal{V}) \quad , \quad \widehat{\mathbf{A}} = \mathcal{W} \cdot_{0,\ldots,d} \mathcal{A}(\mathcal{V}), \tag{2.8}$$

$$\widehat{\mathbf{f}} = (f^T, \mathbf{1}_{N_1}^T, \ldots, \mathbf{1}_{N_d}^T) \cdot_{0,\ldots,d} \mathcal{W} \quad , \quad \widehat{\mathbf{c}} = (c^T, w_1^T, \ldots, w_d^T) \cdot_{0,\ldots,d} \mathcal{V}. \tag{2.9}$$

Let us elaborate on some of the steps in the algorithm. Step 6 is the orthonormalization of $\mathcal{T}_\ell$ against $\mathcal{V}_{:,\ldots,:,1:\ell}$. This can be done using the classical Gram-Schmidt process in tensor form:

$$\begin{aligned} h &= (\mathcal{T}_\ell) \cdot_{0,\ldots,d} \mathcal{V}_{:,\ldots,:,1:\ell} \\ \mathcal{T}_\ell &= \mathcal{T}_\ell - (h^T) \cdot_{d+1} \mathcal{V}_{:,\ldots,:,1:\ell} \\ \mathcal{V}_{:,\ldots,:,\ell+1} &= \mathcal{T}_\ell / \|\mathcal{T}_\ell\|_F. \end{aligned}$$

In step 9 the reduced model is build using tensor contractions. In (2.8), operators $\mathcal{A}$ and $\mathcal{E}$ are first applied to $\mathcal{V}$. Then, contraction with $\mathcal{W}$ in the first $d+1$ modes results in matrices $\widehat{\mathbf{E}}$ and $\widehat{\mathbf{A}} \in \mathbb{C}^{k \times k}$. The contractions (2.9) result in the vectors $\widehat{\mathbf{f}}$ and $\widehat{\mathbf{c}} \in \mathbb{C}^k$.

Calculating the initial tensor $\mathcal{T}_0$ in step 2 and getting the new iteration tensors $\mathcal{T}_\ell$ in Step 5 requires the solution of parametric systems. We use a tensor Krylov method [23][6] to find an approximate solution. In the referred works, it is assumed that the parametric solution can be approximated well by a low rank tensor. If this is not possible, no gain is to be expected from a

tensor Krylov method and model reduction in general. For the technical details, we refer to [23] and the available Matlab software.

With the aim to reduce the number of iterations, we precondition (2.5) and (2.6) to the form

$$A(0)^{-1}\cdot_0\mathcal{A}(\mathcal{T}_0) = A(0)^{-1}\cdot_0(f, \mathbf{1}_{N_1}, \ldots, \mathbf{1}_{N_d}) \tag{2.10}$$

$$A(0)^{-1}\cdot_0\mathcal{A}(\mathcal{T}_\ell) = A(0)^{-1}\cdot_0\mathcal{E}(\mathcal{V}_{:,\ldots,:,\ell}). \tag{2.11}$$

Here, we assume that $\gamma = 0$ corresponds to the mean or expected value of the parameters. This is usually somewhere in the middle of the parameter range. When γ is close to zero, the above preconditioned systems are efficiently solved with a Krylov method. To simplify notation we take $A(0) = A_0$. The theory in the next section can be generalized for $A(0) \neq A_0$. In tensor notation and presuming $A(0) = A_0$, (2.10) and (2.11) become

$$\mathcal{T}_0 + \sum_{j=1}^{d}(A_0^{-1}A_j, \Sigma_j)\cdot_{0,j}\mathcal{T}_0 = (A_0^{-1}f, \mathbf{1}_{n_1}, \ldots, \mathbf{1}_{n_d}) \tag{2.12}$$

$$\mathcal{T}_\ell + \sum_{j=1}^{d}(A_0^{-1}A_j, \Sigma_j)\cdot_{0,j}\mathcal{T}_\ell = (A_0^{-1}E_0)\cdot_0\mathcal{V}_{:,\ldots,:,\ell-1} + \sum_{j=1}^{d}(A_0^{-1}E_j, \Sigma_j)\cdot_{0,j}\mathcal{V}_{:,\ldots,:,\ell-1}. \tag{2.13}$$

3. Moment matching properties. The ℓth moment of the state vector x of system (1.1), in the Laplace variable, is:

$$m_\ell(\gamma) = (A(\gamma)^{-1}E(\gamma))^\ell A(\gamma)^{-1}f \quad , \quad \ell = 0, 1, 2, \ldots, k. \tag{3.1}$$

To make subsequent notation less involved we will mostly use $m_\ell(\gamma) = m_\ell$. Restricting γ to the grid $\mathbb{G}$, we can use tensor notation:

$$\mathcal{M}_\ell = \text{tensor}_{\mathbb{G}}(m_\ell(\gamma)) : (\mathcal{M}_\ell)_{:,I} = m_\ell(\gamma^{(I)}) \quad , \quad \forall I \in \mathbb{I}.$$

The following sections take a closer look at the moments $m_\ell(\gamma)$. We will use the following notation to make the presentation easier:

$$\begin{aligned} S_j &= A_0^{-1}A_j \quad , \quad j = 1, \ldots, d, \\ P_j &= A_0^{-1}E_j \quad , \quad j = 0, \ldots, d, \\ b &= A_0^{-1}f, \end{aligned}$$

and

$$S(\gamma) = I + \sum_{j=0}^{d}\gamma_j S_j,$$

$$P(\gamma) = P_0 + \sum_{j=0}^{d}\gamma_j P_j.$$

We will prove two main observations:

1. there is a connection between multivariate moment matching and tensor Krylov methods (see also Figure 4.1);
2. when the ranks of A_j and E_j for $j = 1, \ldots, d$ are low, there is a connection with block Krylov methods.

3.1. Simplified case. We first consider a special situation, namely $A \equiv A_0$. In this case, the moments can be represented explicitly as a function of γ. From (3.1) we get:

$$\begin{aligned} m_0 &\equiv b \\ m_{\ell+1} &= (P_0 + \sum_{j=1}^{d}\gamma_j P_j)m_\ell \quad , \quad \ell \geq 0. \end{aligned} \tag{3.2}$$

Using these relations, we will find the spaces spanned by the moments. We can now formulate the moment matching theorem for the SISO system for the mean (2.4).

THEOREM 3.1. *If vector w_j and the interpolation points $\gamma_j^{(i)}$, $i = 1, \ldots, N_j$, for $j = 1, \ldots, d$ correspond to a quadrature formula with degree of accuracy at least $2k-1$, then the ℓ-th moment of the output of* (2.2) *around zero matches, for $\ell = 0, \ldots, 2k-1$.*

Proof. The ℓ-th moment of z around zero is

$$\int_\Gamma c^T m_\ell(\gamma) d\gamma.$$

The integrand is a polynomial in $\gamma_1, \ldots, \gamma_d$ of degree ℓ. The ℓ-th moment of z_N of (2.4) is

$$(c^T, w_1^T, \ldots, w_p^T) \cdot \mathcal{M}_\ell = \sum_{(i_1,\ldots,i_p)} (w_1)_{i_1} \cdots (w_d)_{i_p} c^T m_\ell(\gamma^{(i_1,\ldots,ip)}).$$

This can be seen as the application of d quadrature rules in the nodes $\gamma^{(i_1,\ldots,i_p)}$. Since the degree of accuracy is at least $2k-1$, the quadrature rules produce the same sum as the integral, i.e.,

$$(c^T, w_1^T, \ldots, w_d^T) \cdot \mathcal{M}_\ell = \int_\Gamma c^T m_\ell(\gamma) d\gamma,$$

which proves the theorem. ∎

LEMMA 3.2. *The vectorspace spanned by the mode-0 unfolding of $\mathcal{M}_\ell = \text{tensor}_{\mathbb{G}}(m_\ell(\gamma))$, with $m_\ell(\gamma)$ defined by* (3.2)*, has dimension at most $\begin{pmatrix} \ell+d \\ \ell-1 \end{pmatrix}$ and is spanned by the vectors:*

$$P_0^\ell b, (P_0 P_1^{\ell-1} + P_1 P_0 P_1^{\ell-2} + \ldots + P_1^{\ell-1} P_0), \ldots, P_d^\ell b.$$

Similarly, the mode-j vectorspace with $j > 0$ is spanned by the following vectors:

$$\mathbf{1}_{N_j}, \Sigma_j \mathbf{1}_{N_j}, \ldots, \Sigma_j^\ell \mathbf{1}_{N_j} \ ,$$

and has dimension exactly $\min(\ell, N_j)$.

Proof. The mode-0 fibers of $\mathcal{M}_\ell$ are given by

$$\begin{aligned} (\mathcal{M}_0)_{:,i_1,\ldots,i_d} &= b \\ (\mathcal{M}_{\ell+1})_{:,i_1,\ldots,i_d} &= (P_0 + \sum_{j=1}^d \gamma_j^{(i_j)} P_j)^\ell b \quad , \quad i_j = 1, \ldots, N_j, j = 1, \ldots, d. \end{aligned} \tag{3.3}$$

For $\ell = 0$, mode-0 is b. For $\ell = 1$, the first mode is

$$P_0 b + \sum_{i=1} \gamma_i P_i b,$$

which is spanned by the $d+1$ vectors $P_i b$, for $i = 0, \ldots, d$. For $P(\gamma)^{\ell+1} b$, we have to count all terms in $\gamma_1^{i_1} \cdots \gamma_d^{i_d}$ with $i_1 + i_2 + \cdots + i_d = \ell$. This number is $\begin{pmatrix} \ell+d \\ \ell-1 \end{pmatrix}$.

The mode-j fibers of $\mathcal{M}_\ell$ with $j > 0$ can be derived in a similar way. For the rank one tensor $\mathcal{M}_0$ it is clear that all mode-j fibers are proportional to 1_{N_j}.*Let*M_ℓ be a tensor with all mode-j fibers spanned by the vectors $\mathbf{1}_{N_j}, \Sigma_j \mathbf{1}_{N_j}, \ldots, \Sigma_j^\ell \mathbf{1}_{N_j}$. We can write this as the sum of rank one tensors R_i: $\mathcal{M}_\ell = \sum_{i=1}^{r_\ell} R_i$. This gives,

$$\mathcal{M}_{\ell+1} = \sum_{i=1}^{r_\ell} \mathcal{P}(\mathcal{R}_i),$$

with

$$\mathcal{P}(\mathcal{R}_i) = (P_0)\cdot_0 \mathcal{R}_i + \sum_{j=1}^{d} (P_j, \Sigma_j)\cdot_{0,j} \mathcal{R}_i.$$

We deduce that fiber $(\mathcal{M}_{l+1})_{i_0,i_1,\ldots,i_{j-1},:,i_{j+1},\ldots,i_d}$ is a linear combination all mode-j fibers of $\mathcal{R}_i$ and mode-j fibers of $(\Sigma_j)\cdot_j \mathcal{R}_i$. The mode-$j$ vectorspace is a Krylov space with a diagonal matrix with distinct entries. Since all components of $\mathbf{1}_{N_j}$ are nonzero, such a space always has dimension ℓ for $\ell \leq N_j$. This follows from the fact that a polynomial of degree $\ell < N_j$ cannot have N_j distinct zeroes. This proves the lemma. ∎

From the proof of Lemma 3.2 it also follows that the total maximum rank of the space spanned by all the mode-0 unfoldings of $\mathcal{M}_0, \ldots, \mathcal{M}_\ell$ is $\begin{pmatrix} \ell + d + 1 \\ \ell \end{pmatrix}$. This rank grows combinatorially with the number of parameters. However, if the problem has low rank parameter dependencies, this increase can be much lower, as the following lemma shows.

LEMMA 3.3. *Let the column rank of $[E_1, \ldots, E_d]$ be r. Then, the mode-0 unfoldings of $\mathcal{M}_0, \ldots, \mathcal{M}_\ell$ span a vectorspace of dimension at most $1 + l + rl$.*

Proof. When $P_1, \ldots, P_d$ have rank at most r, we have that $P_j = VR_j$ with $V, R_j^T \in \mathbb{C}^{n\times r}$. In this case, we can write (3.3) as

$$(P_0 + VR(\gamma))^\ell b \quad \text{with} \quad R(\gamma) = \sum_{j=1}^{d} \gamma_j R_j \ ,$$

with $\gamma = \gamma^{(i_1,\ldots,i_d)}$. The mode-0 unfoldings span a block Krylov space

$$[b, V], P_0[b, V], \ldots, P_0^{\ell-1}[b, V], P_0^\ell b \ ,$$

which proves the lemma. ∎

Lemma 3.3 shows that for low rank parametric dependencies, the mode-0 vectorspace of the moment tensors can also be generated by a block Krylov method, as in [39].

3.2. General case. In this section, we consider the more general situation. When $A \not\equiv A_0$, it is not feasible to compute or represent the moments m_ℓ exactly. The moments are now represented by the following recursive relation:

$$(I + \sum_{j=1}^{d} \gamma_j S_j) m_0 = b \tag{3.4}$$

$$(I + \sum_{j=1}^{d} \gamma_j S_j) m_{\ell+1} = (P_0 + \sum_{j=1}^{d} \gamma_j P_j) m_\ell. \tag{3.5}$$

Each moment is the solution of a preconditioned parametric system, which is solved using a tensor Krylov method. Instead of representing the exact moments m_ℓ, we use the approximate solutions of the parametric systems, $\widetilde{m}_\ell$, as a result of the Krylov method. The accuracy of the solution depends on the specific iterative method used. We focus on Richardson iteration. This particular Krylov method is a relevant case, since the subspace built by any Krylov method is using the same sequence of vectors.

LEMMA 3.4. *When Richardson iteration with zero initial solution is used for solving* (3.4)*, the solution takes the form*

$$\widetilde{m}_0 = \sum_{i=0}^{\nu} (-\sum_{j=1}^{d} \gamma_j S_j)^i b$$

Proof. The solution is obtained through the following iterative process:

- $\widetilde{m}_0^{(0)} = 0$
- For $i = 0, \ldots, \nu - 1$:

$$\widetilde{m}_0^{(i+1)} = \widetilde{m}_0^{(i)} + (b - S(\gamma)\widetilde{m}_0^{(i)})$$

Applying this recursively, taking into account that $I - S = -\sum_{j=1}^{d} \gamma_j S_j$ leads to the proof of the lemma. □

LEMMA 3.5. *When Richardson iteration with zero initial solution is used for solving* (3.5)*, the solution takes the form*

$$\widetilde{m}_{\ell+1} = \sum_{i=0}^{\nu} (-\sum_{j=1}^{d} \gamma_j S_j)^i P(\gamma) \widetilde{m}_\ell$$

Proof. The proof is similar to the one of Lemma 3.4. □

LEMMA 3.6. *With $S_j = A_0^{-1} A_j$, $j = 1, \ldots, d$, we have that*

$$(A_0 + \sum_{j=1}^{d} \gamma_j A_j)^{-1} = \sum_{i=0}^{\nu} (-\sum_{j=1}^{d} \gamma_j S_j)^i A_0^{-1} + O(\| \sum_{j=1}^{d} \gamma_j S_j \|^{\nu+1})$$

Proof. One can find $(A_0 + \sum_{j=1}^{d} \gamma_j A_j)^{-1}$ by solving:

$$(A_0 + \sum_{j=1}^{d} \gamma_j A_j) x(\gamma) = I,$$

or, after preconditioning,

$$(I + \sum_{j=1}^{d} \gamma_j S_j) x(\gamma) = A^{-1}.$$

Using Lemma 3.4, a power series for the solution is:

$$x(\gamma) = \sum_{i=0}^{\infty} (-\sum_{j=1}^{d} \gamma_j S_j)^i A_0^{-1}.$$

Cutting off the power series at $i = \nu$, leads to an error $O(\| \sum_{j=1}^{d} \gamma_j S_j \|^{\nu+1})$, proving the lemma. □

Notice that for Richardson iteration, to converge to the correct solution, we should have that

$$\| \sum_{j=1}^{d} \gamma_j S_j \| < 1.$$

Because we use the approximate solutions $\widetilde{m}_\ell$, the moments of (2.4) are no longer exactly matched. Using the previous lemma's we can make a statement about the difference between the matched moments, when Richardson iterations are used to solve the parametric systems.

THEOREM 3.7. *Let m_ℓ, $\ell = 0, \ldots, k$ be the exact moments around zero of the state vector of* (2.2)*. Let $\widetilde{m}_\ell$, $\ell = 0, \ldots, k$ be the moments around zero of the state vector of* (2.2) *where the linear system solves are performed by ν preconditioned Richardson iterations. Then*

$$\| \widetilde{m}_\ell - m_\ell \| = O(\| \sum_{j=1}^{d} \gamma_j S_j \|^{\nu+1}) \text{ for } \ell \geq 0 \text{ and } \gamma \in \mathbb{G}.$$

Proof. The exact moments are computed from the Krylov space with starting vector $A(\gamma)^{-1}f$ and matrix $S(\gamma)^{-1}T(\gamma)$. Richardson iteration computes the solution using the matrix

$$R(\gamma) = \sum_{i=0}^{\nu}(-\sum_{j=1}^{d}\gamma_j S_j)^i P(\gamma), \tag{3.6}$$

instead of $S(\gamma)^{-1}P(\gamma)$, and starting vector

$$\widetilde{m}_0 = \sum_{i=0}^{\nu}(-\sum_{j=1}^{d}\gamma_j S_j)^i b,$$

instead of $S(\gamma)^{-1}b$. Since $R(\gamma)$ is independent of the iteration count, the moments are computed using the recurrence

$$\widetilde{m}_{\ell+1} = R(\gamma)\widetilde{m}_\ell \quad , \quad \ell \geq 0.$$

For the initial vector, it follows from Lemma 3.6 that

$$\widetilde{m}_0 - S(\gamma)^{-1}b = O(\|\sum_{j=1}^{d}\gamma_j S_j\|)^{\nu+1}. \tag{3.7}$$

For the other moments, we use an induction argument on ℓ. Assume the theorem holds for $\widetilde{m}_{\ell-1}$. We know that

$$\begin{aligned}\|\widetilde{m}_\ell - m_\ell\| &\leq \|\widetilde{m}_\ell - R(\gamma)m_{\ell-1}\| + \|R(\gamma)m_{\ell-1} - m_\ell\| \\ &= \|R(\gamma)(\widetilde{m}_{\ell-1} - m_{\ell-1})\| + \|(R(\gamma) - S(\gamma)^{-1}P(\gamma))m_{\ell-1}\|.\end{aligned}$$

Both terms in the right-hand side are $O(\|\sum_{j=1}^{d}\gamma_j S_j\|^{\nu+1})$, which proves the induction step. ∎

When other Krylov methods are used to solve the parametric systems, the above theorem no longer holds. We know, however, that the solution for Krylov methods will be equal to

$$\widetilde{t}_j = \sum_{i=0}^{\nu}\zeta_i(-\sum_{j=1}^{d}\gamma_j S_j)^i P\widetilde{v}_j,$$

where ζ_i are scalar constants. From this, it follows that all Krylov methods result in the same mode-0 vector space of the moments, as stated in the following theorem.

THEOREM 3.8. *The vector space spanned by the mode-0 unfoldings of $\mathcal{M}_0, \ldots, \mathcal{M}_\ell$, with $\mathcal{M}_i = tensor_{\mathbb{G}}(m_i)$ as defined by* (3.4) *and* (3.5)*, is spanned by the coefficients of the following polynomials:*

$$\begin{gathered}S(\gamma)^\nu b, \\ S(\gamma)^\nu P(\gamma)S(\gamma)^\nu b, \ldots \\ (S(\gamma)^\nu P(\gamma))^{k-1}S(\gamma)^\nu b\end{gathered}$$

Proof. The above polynomials are the iteration vectors obtained after ν iterations of a Krylov method applied to $S(\gamma)$, with starting vector respectively b for the first iteration and $\widetilde{v}_j$ for the remaining iterations. ∎

From Theorem 3.8, one can see that an upper bound for the dimension of the space spanned by mode-0 unfoldings of the moments $\mathcal{M}_0, \ldots, \mathcal{M}_\ell$ is equal to $\begin{pmatrix}(\ell-1)\nu+\ell+d+1 \\ (\ell-1)\nu+\ell\end{pmatrix}$. So, as in the simplified case, the maximum rank increases combinatorially with the number of parameters but faster then in the simplified case. We can, however, again state a better upper bound for the dimension of this space, in case the matrices A_j and E_j, $j = 1, \ldots, d$ are of low rank.

THEOREM 3.9. *Assume that the column rank of* $[A_j, E_j, j = 1, \ldots, d]$ *is* r*. Then, there are* L*,* U*,* $\widetilde{S}_j$*, and* $\widetilde{P}_j$*,* $j = 1, \ldots, d$ *so that* $S_j = L\widetilde{S}_j U^*$ *and* $P_j = L\widetilde{P}_j U^*$ *with* $L, U \in \mathrm{R}^{n\times r}$ *for* $j = 1, \ldots, d$*. Then, the mode-0 unfolding of* $\mathcal{M}_\ell$ *is in* $\mathrm{Range}(V_\ell)$ *with* $V_\ell = [[b, L], P_0[b, L], \ldots, P_0^{\ell-1}[b, L]]$.

Proof. From the definition of m_ℓ, we derive that

$$\begin{aligned} m_0 &= b - \sum_{j=1}^{d} \gamma_j S_j m_0 \\ &= b - L\left(\sum_{j=1}^{d} \gamma_j \widetilde{S}_j U^* m_0\right), \end{aligned}$$

which is spanned by $[b, L]$ and has rank at most $r + 1$. Further, we have that

$$\begin{aligned} m_1 &= P_0 m_0 + \sum_{j=1}^{d} \gamma_j P_j m_0 - \sum_{j=1}^{d} \gamma_j S_j m_1 \\ &= P_0 m_0 + L\left(\sum_{j=1}^{d} \gamma_j (\widetilde{P}_j U^* m_0 - \widetilde{S}_j m_1)\right). \end{aligned}$$

The first term in the last line, is spanned by $P_0[b, L]$, where the second term is spanned by $[b, L]$. By induction, we see that

$$\begin{aligned} m_{\ell+1} &= -P_0 V_\ell \widehat{m}_\ell - \sum_{j=1}^{d} P_j V_\ell \widehat{m}_\ell + \sum_{j=1}^{d} S_j m_{\ell+1} \\ &= -P_0 V_\ell \widehat{m}_\ell - L\sum_{j=1}^{d} (\widetilde{P}_j U^* V_\ell \widehat{m}_\ell + \widetilde{S}_j U^* m_{\ell+1}), \end{aligned}$$

which is spanned by the columns of $[P_0^\ell[b, L], P_0^{\ell-1}[b, L], \ldots, P_0[b, L], [b, L]]$. This proves the theorem. ∎

As in the simplified case, the rank of the mode-0 vector space of the moments, could remain low in case of low-rank parameter dependencies. The result of this theorem is also found in [39].

We have presumed that using tensors is relevant as long as they can be represented in a low rank format. We have seen that if the problem has a low rank parameter dependency, the tensor rank grows slowly. If the rank of the matrices A_i and E_i is not low, one can still truncate the tensors by discarding small singular values. If the required level of accuracy is not too stringent, the rank of the moments could remain low as well. This will be further investigated in the section on numerical examples.

4. Parametric reduced models. In the previous sections we introced a system that outputs the mean over the parameter range. We can compare this to parametric reduced models, which is typically the result of classical parametric model reduction. We can use the basis tensors $\mathcal{V}$ and $\mathcal{W}$ to set this up. We use the mode-0 unfoldings of these tensors:

$$\begin{aligned} \mathcal{V} &= (V_0)\cdot_0 \widetilde{\mathcal{V}} \quad , \quad \widetilde{\mathcal{V}} = (V_0^*)\cdot_0 \mathcal{V} \\ \mathcal{W} &= (W_0)\cdot_0 \widetilde{\mathcal{W}} \quad , \quad \widetilde{\mathcal{W}} = (W_0^*)\cdot_0 \mathcal{W} \ . \end{aligned} \tag{4.1}$$

The reduced problem becomes:

$$\begin{aligned} \widehat{E}\dot{x} + \widehat{A}x &= \widehat{f}u(t) \\ \widehat{y} &= \widehat{c}^T x \\ \widehat{z}_N &= \sum_{i_1=1,\ldots,N_1} \cdots \sum_{i_d=1,\ldots,N_d} w_1^{(i_1)} \cdots w_d^{(i_d)} \widehat{y}(\gamma^{(i_1,\ldots,i_d)}), \end{aligned} \tag{4.2}$$

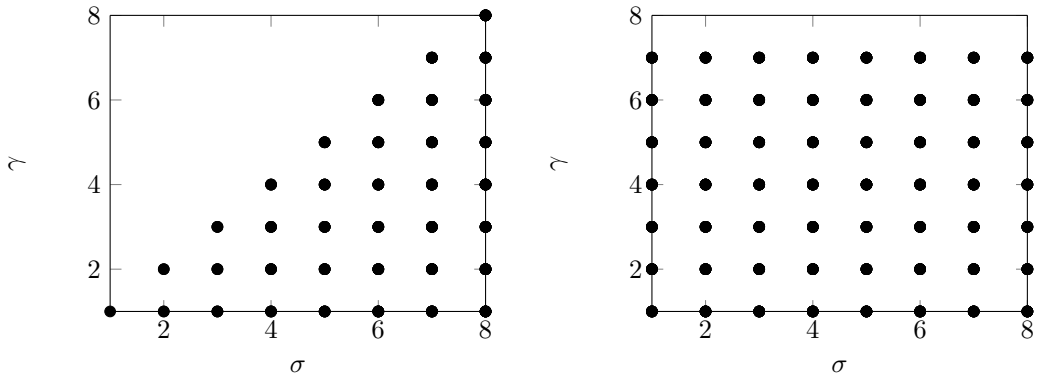

Fig. 4.1: Moments matched by the parametric reduced model. (left) For the simplified case $\mathcal{A} \equiv A_0$. (right) For the general case $\mathcal{A} \not\equiv A_0$.

with

$$\begin{aligned}
\widehat{E} &= W_0^* E_0 V_0 + \sum_{j=1}^{d} \gamma_j W_0^* E_j V_0 \\
\widehat{A} &= W_0^* A_0 V_0 + \sum_{j=1}^{d} \gamma_j W_0^* A_j V_0 \\
\widehat{f} &= V_0^* f \\
\widehat{c} &= W_0^* c.
\end{aligned}$$

So, using the mode-0 unfoldings of iteration tensors $\mathcal{V}$ and $\mathcal{W}$, a reduced parametric model is set up. As with multivariate Padé methods, moments are matched in the parameter space as well as for the Laplace variable. For the simplified case, moments up to a degree ℓ will be matched for the parameters and the Laplace variable. For the general case, the coefficients from Theorem 3.8 are used, where moments up to degree k are matched for the Laplace variable and up to a degree ν for the parameters. We have that:

$$\frac{\partial^i}{\partial s^i}\frac{\partial^{i_1}}{\partial \gamma_1^{i_1}}\cdots\frac{\partial^{i_d}}{\partial \gamma_1^{i_d}} y = \frac{\partial^i}{\partial s^i}\frac{\partial^{i_1}}{\partial \gamma_1^{i_1}}\cdots\frac{\partial^{i_d}}{\partial \gamma_1^{i_d}} \widehat{y} \quad , \quad i = 0, \ldots, k, 0 \leq \sum_{j=1}^{d} i_j \leq \nu.$$

Both situations are compared in Figure 4.1. In fact, it can be seen that the tensor Krylov method, as presented in this paper, combines moment matching with interpolation in the parameter space. From Theorem 3.7 it follows that if ν is large, for every parameter on the grid, each of the Laplace variable moments are accurately determined. This can be seen as interpolation for every parameter on the grid. On the other hand, if ν is low, only few moments for parameters on the grid are matched, as is the case with multivariate moment matching techniques.

The parametric reduced model (4.2) is much smaller than the original model (1.1) and produces more accurate z than (2.7) of the same size. On the other hand, it requires the computation of z_N by numerical integration over the parameter range, which can be expensive. The parametric reduced model can be seen as a projection of (2.4) on the subspace $\text{Range}(I_N \otimes V_0)$ of dimension Nk. Because this subspace is N times the size of the reduced model (2.3), this leads to a much more accurate solution, but it is also more expensive to evaluate. In §5, we will compare the quality of (2.7) and (4.2). One could also try to combine both approaches by using an intermediate reduced parametric model to set up a reduced model for the mean. This is summarized in the following algorithm:

ALGORITHM 4.1 (Two level approach).

1: Use Algorithm 2.2 to build a reduced model of order k.
2: Use W_0 and V_0 to build (4.2).
3: Apply Algorithm 2.2 to build a reduced model of order $\widetilde{k}$ for system (4.2).

5. Numerical examples.

5.1. Implementation details. We used a hierarchical form of the Tucker format to store the tensors. This was implemented in Matlab based on the H-Tucker library [24]. Another possibility would have been to use tensor trains, a special case of hierarchical Tucker. It should be noted that the library is a great research tool, but performance is still not good enough for solving real applications. For the purpose of this paper, this implementation is very suitable.

A key issue in the methods presented in this paper is the rank reduction of tensors at various stages of Algorithm 2.2. In particular, rank reduction is carried out at the end of steps 2, 5 and 6. The rank reduction is based on a higher order singular value decomposition of the resulting tensor and then truncation to low rank by dropping the singular values below the relative tolerance τ.

5.2. Wave equation. Consider the damped wave equation inside the 3D unit box, with parametric Robin boundary conditions on two faces:

$$(K + i\omega C(\gamma) - \omega^2 M)x = f \quad \text{with} \quad C(\gamma) = \gamma_1 C_1 + \gamma_2 C_2$$
$$y = c^T x.$$

K is the discretized Laplacian operator, $C(\gamma)$ is the damping matrix and M is the massmatrix. This model uses a discretization by finite differences and has a dimension $n = 1000$. The parameters $\gamma_{1,2}$ are uniformly distributed on the interval $[0.2, 0.9]$. We used four Legendre quadrature points for each parameter. The problem is reformulated as the linear problem:

$$\left(\begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} + i\omega \begin{bmatrix} C(\gamma) & M \\ -I & 0 \end{bmatrix} \right) \begin{pmatrix} x \\ i\omega x \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix},$$

which defines $A, E \in \mathbb{R}^{2n \times 2n}$. Note that A does not depend on γ, which means that we do not require a tensor Krylov method for solving the linear systems in steps 2 and 5 of Algorithm 2.2 (the simplified case).

We compare the reduced model for the mean (2.7) with the parametric reduced model (4.2). Figures 5.1 and 5.2 show, for different values of the truncation error τ, the relative error between the output of the full model and the reduced models. Figure 5.1 displays the error after 10 Krylov iterations and Figure 5.2 after 30. Table 5.1 shows the mode-0 rank of $\mathcal{V}_\ell$, compared to the maximal rank from Lemma 3.2. Table 5.2 shows computational times.

The following conclusions can be drawn. Small truncation errors τ quickly lead to large ranks of the tensors, and thus high orders of the parametric reduced model, during the first few steps of the Arnoldi process. As expected, the parametric model is more accurate than the model for the mean. However, the computational time to solve the parametric reduced models is significantly higher than that of the reduced models for the mean. The quality of the mean computed from (2.7) improves when τ is smaller, but because of the growing rank at a significantly higher cost. In contrast, the quality of (4.2) does not significantly improve with decreasing τ[1].

The fact that lowering the tolerance leaves the parametric model relatively unaffected pleads for the two-level approach presented in Algorithm 4.1. The large scale problem is first reduced to a parametric reduced model using few Arnoldi steps with a relatively large value of τ. A reduced model for the mean can subsequently be built from this parametric model using more steps and a smaller τ. Figure 5.3 shows the error for a hybrid case. It shows the error of a mean model set up by applying Arnoldi steps to the full model and of a mean model based on a parametric reduced model that was set up using 10 Arnoldi steps and a tolerance $\tau = 10^{-8}$. Both errors are

[1] For $\tau = 10^{-4}$, truncation of the matrix of iteration vectors (stored as tensor $\mathcal{V}_{:,:,\ldots,1:l}$) causes a larger error. If we store every iteration tensor $\mathcal{V}_{:,:,\ldots,i}$ separately instead of in one (truncated) tensor, the accuracy is comparable to that of lower truncation tolerances, but more memory will be needed.

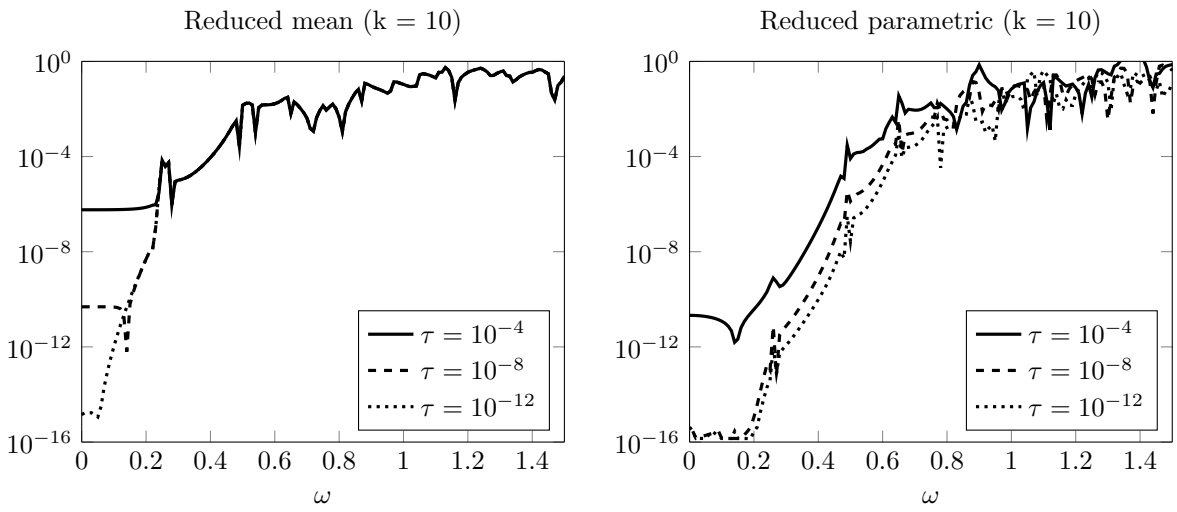

Fig. 5.1: Relative error on the output of the reduced model for $k = 10$. (left) Model (2.7), (right) Model (4.2).

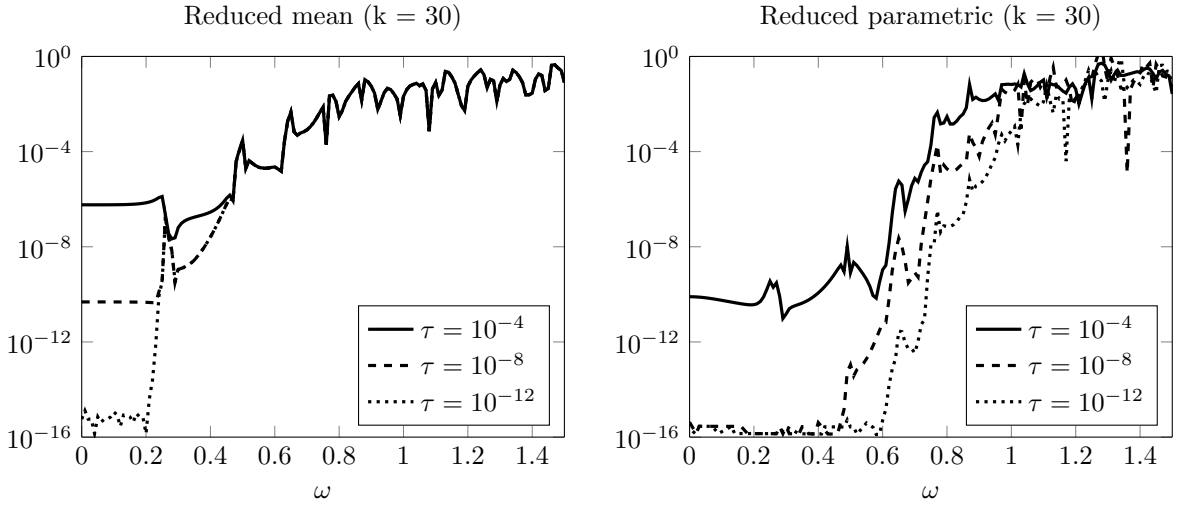

Fig. 5.2: Relative error on the output of the reduced model for $k = 30$. (left) Model (2.7), (right) model (4.2).

comparable while the time to calculate the reduced model in the mixed case is almost halved (0.11 instead of 0.21 times the full calculation time).

5.3. Footbridge problem with eight parameters. In this example, we study the footbridge located over the Dijle river in Mechelen, Belgium (see the sketch in Figure 5.4). It is about 31.354 meters in length and four tuned mass dampers (TMDs) are located at main nodes corresponding to 11.299m, 19.314m, 10.549m and 20.309m respectively, each of which are 40.72kg in weight.

Table 5.1: Mode-0 rank of $\mathcal{V}_\ell$ for different values of τ.

ℓ	$\tau = 10^{-4}$	$\tau = 10^{-8}$	$\tau = 10^{-12}$	Lemma 3.2
0	1	1	1	1
1	4	4	4	4
2	7	10	10	10
3	9	19	20	20
4	12	26	33	35
5	14	32	46	56
6	17	39	58	84
7	19	46	71	120
8	20	51	83	165
9	22	55	94	220
10	23	60	103	286
11	27	65	111	364
⋮				
20	41	95	162	
21	42	98	167	
⋮				
29	49	118	203	
30	50	121	207	

Table 5.2: Computational time for the wave equation problem using $\tau = 10^{-8}$ (expressed as fraction of time for solving the full problem).

	k = 10	k = 20	k = 30
Offline	0.025	0.067	0.12
Online (parametric)	0.047	0.081	0.12
Online (mean)	0.00025	0.00041	0.00046

The discretized model which is describing the footbridge dynamical system is

$$\begin{cases} \Big(K_0 + i\omega C_0 + \sum_{i=1}^{4}(k_i + i\omega c_i)K_i - \omega^2 M_0\Big)x = f, \\ y = c^T x, \end{cases} \tag{5.1}$$

where K_0 and M_0 are obtained from a finite element model with $n = 25962$ degrees of freedom (DOFs) and $C_0 = 0.1003M_0 + 0.0001591K_0$. K_i are rank one matrices that represent the interaction between the i-th TMD and the footbridge. The input vector f represents a unit excitation at the central span and the output vector c picks out the displacement at the central span. Since the order of magnitude of the objective function is too small compared with those of the design parameters , we set $c = 10^6 c$ to avoid numerical difficulties.

The frequency range of interest is $[\omega_L, \omega_H] = [0\,{}^{\text{rad}}/_{\text{s}}, 10\pi\,{}^{\text{rad}}/_{\text{s}}]$, [37]. The stiffness k_i and damping c_i coefficients are the 8 parameters of the system (5.1), so $\gamma^{(j)} = (c_1, c_2, c_3, c_4, k_1, k_2, k_3, k_4)$. Similar to the previous example, we use the lattice rule [9] with four interpolation points, which in this case are the points shown in Table 5.3. The last column of the table gives the range of the parameter corresponding to that row. The parametric domain is discretized by a Gauss-Legendre

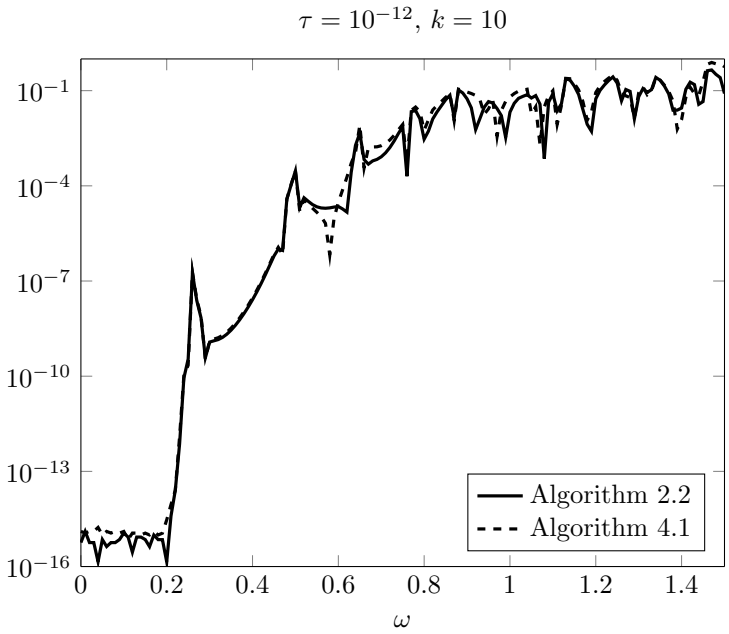

Fig. 5.3: Relative error for a mean model ($\tau = 10^{-12}$, $k = 30$) set up using the full model (Algorithm 2.2) and set up using a reduced parametric model with $\tau = 10^{-8}$ and $k = 10$ (Algorithm 4.1).

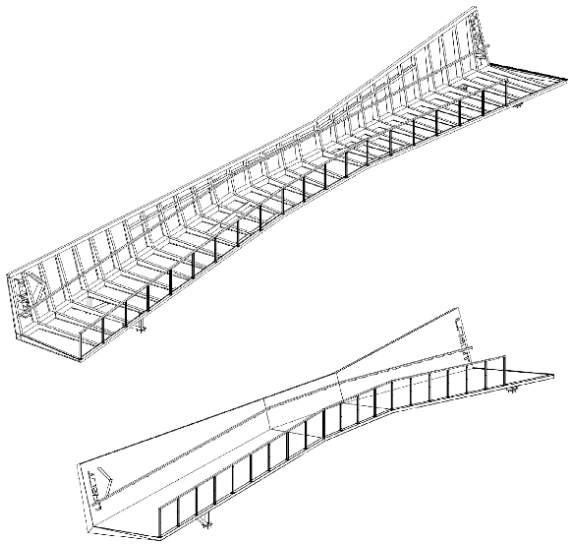

Fig. 5.4: The conceptual model of the footbridge.

quadrature rule with four points. As in the previous example, the problem is reformulated:

$$\left(\begin{bmatrix} K_0 & C_0 \\ 0 & I \end{bmatrix} + \sum_{i=1}^{4} k_i \begin{bmatrix} K_i & 0 \\ 0 & 0 \end{bmatrix} + i\omega \sum_{i=1}^{4} c_i \begin{bmatrix} K_i & 0 \\ 0 & 0 \end{bmatrix} + i\omega \begin{bmatrix} 0 & M_0 \\ -I & 0 \end{bmatrix} \right) \begin{pmatrix} x \\ i\omega x \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix},$$

which defines $A, E \in \mathbb{R}^{2n \times 2n}$. Notice that A now is function of γ so that we have to solve a parametric system in every iteration step. To do so, we use a tensor GMRES method with two

Table 5.3: The interpolation points for the example of footbridge damper are found by lattice rule parameter $\gamma^i = (c_1, c_2, c_3, c_4, k_1, k_2, k_3, k_4)$.

	$\gamma^{(1)}$	$\gamma^{(2)}$	$\gamma^{(3)}$	$\gamma^{(4)}$	desired interval
c_1	40	50	45	55	$[40, 60]$
c_2	27	37	32	42	$[27, 47]$
c_3	55	45	50	40	$[35, 55]$
c_4	23	33	38	28	$[23, 43]$
k_1	20000	25000	27500	22500	$[20000, 30000]$
k_2	16000	21000	18500	23500	$[16000, 26000]$
k_3	18000	23000	20500	25500	$[18000, 28000]$
k_4	14000	19000	16500	21500	$[14000, 24000]$

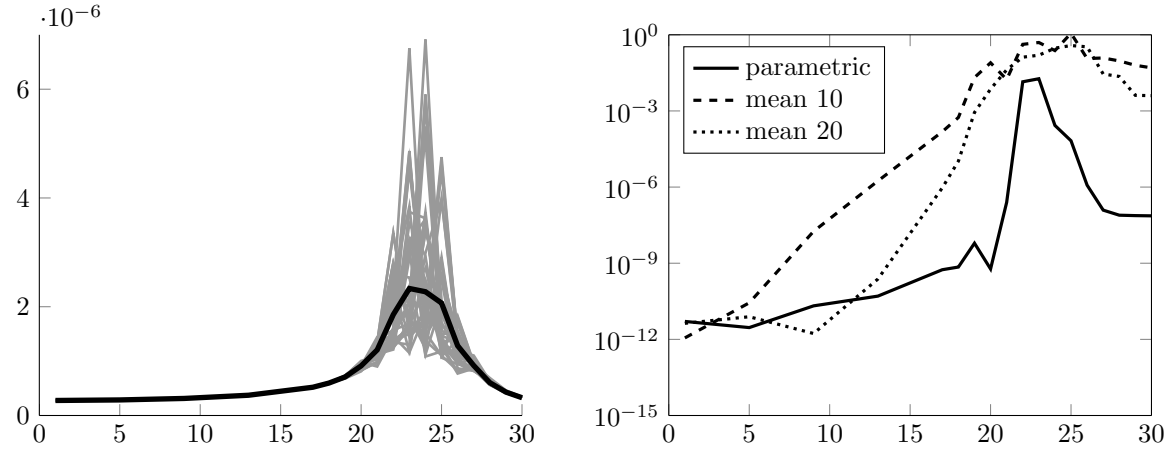

Fig. 5.5: (left) Bundle of parametric solutions and the mean computed by tensorization of the parameter space, (right) relative error for a parametric reduced system ($\tau = 10^{-8}$, $k = 10$) and for two mean reduced systems ($\tau = 10^{-12}$, $k = 10$ and $k = 20$).

restarts and five steps per restart. Also, Theorem 3.9 can be applied here with $[K_1, K_2, K_3, K_4]$ having rank 4.

Figure 5.5 shows the mean solution together with a bundle of parametric solutions. It also shows the error for the parametric reduced model and for two mean systems. Table 5.4 gives the computational times.

The rank of the mode-0 unfoldings of the iteration vectors are given in Table 5.5. Notice that the rank of the zero mode, in case of $\tau = 10^{-12}$, increases above the theoretical maximum value of Theorem 3.9. We did not find the reason why the rank increased above the theoretical upper bound, but we think that the requested tolerance is below the level of rounding errors in some of the operations on the data structure. This is subject of further research. We also noticed that the internal ranks of the hierchical Tucker data structure can become much larger then the rank of the mode-0 unfoldings. This can cause large computational times, as for example is apparent in the offline calculation time for the base after 20 Arnoldi steps that amounts to 0.6 times the time for solving the full problem. This is also the case setting up the base for the mean reduced model (Algorithm 4.1), making this no longer a profitable solution. These points highlight the problem with the use of the hierarchical Tucker tensor format in combination with tensor Krylov methods while solving difficult problems like the footbridge problem.

Table 5.4: Computational time for bridge problem using $\tau = 10^{-8}$ (expressed as fraction of time for solving the full problem).

	k = 5	k = 10	k = 20
offline	0.0074	0.086	0.60
online (parametric)	0.057	0.080	0.089
online (mean)	1.1×10^{-7}	1.9×10^{-7}	11×10^{-7}

Table 5.5: Rank of V_0 for the footbridge problem.

j	$\tau = 10^{-4}$	$\tau = 10^{-8}$	$\tau = 10^{-12}$	Theorem 3.9
0	1	1	1	1
1	2	2	3	6
2	7	7	8	11
3	12	12	13	16
4	14	17	18	21
5	17	22	23	26
6	22	29	29	31
7	23	34	34	36
8	26	40	43	41
9	27	45	48	46
10	28	51	62	51
⋮				
19	33	92	157	96
20	34	91	154	101

6. Conclusions. Given a model that depends on a set of stochastically uncorrelated parameters, we presented a SISO system for approximating the mean of its output. This was done using quadrature rules after choosing interpollation points on a cartesian grid in parameter space. The resulting SISO system matches moments for the mean up to a certain degree that depends on the accuracy of the quadrature formula.

Furthermore, we used tensors to represent the state variables for all parameters and we used a two-sided Arnoldi process to reduce the SISO system for the mean. Using tensors for model reduction is useful in case the tensors can be represented in a low rank form. We proved this to be the case when there is a low rank parametric dependency in the problem. Otherwise, the tensors can be truncated to low dimensional form, if the required level of accuracy allows for it. However, the use of the hierarchical Tucker format resulted in some cases in high internal ranks that cause increased computational times.

Using the tensor Arnoldi method, moments for the mean are only matched within a certain tolerance, as in every step the solution of a parametric system is approximated by a tensor Krylov method. Moments are matched for the Laplace variable as well as for the parameters as is the case with multivariate moment matching. Using a large amount of tensor Krylov steps to solve the parametric system, one accurately determines the transfer function for the parameters on the grid, resembling interpolatory reduced models in the parameter space. Therefore, the tensor reduction technique, as presented in this paper, is interesting because of it's hybrid form between multivariate Padé methods and interpolatory model reduction.

Acknowledgements. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State Science Policy Office. The scientific responsibility rests

with its author. The research is also partially funded by the KU Leuven Research Council grants PFV/10/002, OT/10/038 and OT/14/074.

REFERENCES

[1] A. C. Antoulas and D. C. Sorensen. Lyapunov, Lanczos, and inertia. Technical Report TR00-13, Rice University, Departement of Computational and Applied Mathematics, Texas, USA, 2000.

[2] A.C. Antoulas, C.A. Beattie, and S. Gugercin. *Interpolatory model reduction of large-scale dynamical systems*, page 56. Springer-Verlag, 2010.

[3] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Structured Matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing, Contemporary Mathematics, AMS publications*, 280:193–219, 2001.

[4] A.C. Antoulas, D.C. Sorensen, and Y. Zhou. On the decay rate of Hankel singular values and related issues. *Systems & Control Letters*, 46:323–342, 2002.

[5] Z. Bai and R. W. Freund. A partial Padé-via-Lanczos method for reduced-order modeling. *Linear Alg. Appl.*, 332–334:139–164, 2001.

[6] B. Beckermann, D. Kressner, and C. Tobler. An error analysis of Galerkin projection methods for linear systems with tensor product structure. 2014. Submitted for publication.

[7] J.-R. Benner. P., Li and T. Penzl. Numerical solution of large-scale lyapunov equations, riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.

[8] A. Bodendiek. *Moment matching based model order reduction in computational electromagnetism.* Phd thesis, TU Braunschweig, Braunscheig, Germany, 2013.

[9] R. Cools, F.Y. Kuo, and D. Nuyens. Constructing embedded lattice rules for multivariate integration. *SIAM J. Sci. Comput.*, 28:2162–2188, 2006.

[10] L. Daniel, O.C. Siong, L.S. Chay, K.H. Lee, and J. White. A multiparameter moment matching model reduction approach for generating geometrically parameterized interconnect performance models. *Trans. on Computer Aided Design of Integrated Circuits*, 23(5):678–93, 2004.

[11] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 24:1253–1278, 2000.

[12] V. de Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[13] L. Eldén and B. Savas. A Newton–Grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor. *SIAM Journal on Matrix Analysis and Applications*, 31(2):248–271, 2009.

[14] P. Feldman and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Computer-Aided Design*, CAD-14:639–649, 1995.

[15] L. Feng. Parameter independent model order reduction. *Mathematics and Computers in Simulation*, 68:221–234, 2005.

[16] L. Feng. Parametric model reduction for systems with coupled parameters. In *Proceedings of the 15th ILAS Conference, Cancún, México*, pages 22–24, 2008.

[17] L. Feng, E. Rudnyi, and Jan Korvink. Preserving the film coefficient as a parameter in the compact thermal model for fast electrothermal simulation. *IEEE Trans. Computer-Aided Design*, 24(12):1838–1847, 2005.

[18] E. Grimme, D. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. *Numerical Algorithms*, 12:1–31, 1996.

[19] S. Gugercin. An iterative SVD-Krylov based method for model reduction of large-scale dynamical systems. *Linear Alg. Appl.*, 428:1964–1986, 2008.

[20] S. Gugercin, A. Antoulas, and C. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM Journal on Matrix Analysis and Applications*, 30:609–638, 2008.

[21] S. Gugercin, D.C. Sorensen, and A.C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Numerical Algorithms*, 32(1):27–55, 2003.

[22] P. Gunupudi and M. Nakhla. Multi-dimensional model reduction of VLSI interconnects. In *Custom Integrated Circuits Conference, 2000. Proceedings of the IEEE 2000*, pages 499–502, 2000.

[23] D. Kressner and C. Tobler. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1288–1316, 2011.

[24] D. Kressner and C. Tobler. htucker: A Matlab toolbox for tensors in hierarchical Tucker format, 2012. http://anchp.epfl.ch/op/edit/page-78133.html.

[25] X. Li, P. Li, and L.T. Pileggi. Parameterized interconnect order reduction with explicit-and-implicit multi-parameter moment matching for inter/intra-die variations. In *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 805–811, 2005.

[26] Y.-T. Li, Z. Bai, and Y. Su. A two-directional Arnoldi process and its application to parametric model order reduction. *Journal of Computational and Applied Mathematics*, 226:10–21, 2009.

[27] Y.-T. Li, Z. Bai, Y. Su, and X. Zeng. Parameterized model order reduction via a two-directional Arnoldi process. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, pages 868–873, 2007.

[28] Y.-T. Li, Z. Bai, Y. Su, and X. Zeng. Model order reduction of parameterized interconnect networks via a Two-Directional Arnoldi process. *Computer-Aided Design of Integrated Circuits and Systems, IEEE*

Transactions on, 27(9):1571–1582, 2008.

[29] E. Luna-Ortiz and C. Theodoropoulos. An input/output model reduction-based optimization scheme for large-scale systems. *Multiscale Modeling and Simulation*, 4(2):691–708, 2005.

[30] L. Machiels, Y. Maday, I.B. Oliveira, A.T. Patera, and D.V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 331(2):153–158, 2000.

[31] B. Peherstorfer, S. Zimmer, and H.-J. Bungartz. *Model Reduction with the Reduced Basis Method and Sparse Grids*. Lecture Notes in Computational Science and Engineering. Springer, Heidelberg, New York, Dordrecht, London, 2013.

[32] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Systems Control Lett.*, 40:139–144, 2000.

[33] J. Rommes and N. Martins. Efficient computation of transfer function dominant poles using subspace acceleration. *IEEE Transactions on Power Systems*, 21(3):1218–1226, 2006.

[34] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15:229–275, 2007.

[35] M. Saadvandi, K. Meerbergen, and W. Desmet. Parametric dominant pole algorithm for parametric model order reduction. *Journal of Computational and Applied Mathematics*, 295:259–280, 2014.

[36] V. Simoncini and E. Gallopoulos. Transfer functions and resolvent norm approximation of large matrices. *ETNA*, 7:190–201, 1998.

[37] R. Van Beeumen, K. Van Nimmen, G. Lombaert, and K. Meerbergen. Model reduction for dynamical systems with quadratic output. *International Journal of Numerical Methods in Engineering*, 91(3):229–248, 2012.

[38] D. S. Weile, Michielssen E., E. Grimme, and K. Gallivan. A method for generating rational interpolant reduced order models of two-parameter linear systems. *Appl. Math. Let.*, 12:93–102, 1999.

[39] Y. Yue and K. Meerbergen. Parametric model order reduction of damped mechanical systems via the block Arnoldi process. *Applied Mathematics Letters*, 26:643–648, 2013.